Using statistical-model-checking-based simulation for evaluating the robustness of a production schedule

Sara Himmiche, Alexis Aubry, Pascale Marangé, Jean-François Pétin, Marie Duflot

▶ **To cite this version:**

Sara Himmiche, Alexis Aubry, Pascale Marangé, Jean-François Pétin, Marie Duflot. Using statistical-model-checking-based simulation for evaluating the robustness of a production schedule. 7th Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing, SOHOMA'17, Oct 2017, Nantes, France. hal-01652140

HAL Id: hal-01652140
https://inria.hal.science/hal-01652140

Submitted on 29 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Using Statistical-Model-Checking-Based Simulation for Evaluating the Robustness of a Production Schedule

Sara Himmiche, Alexis Aubry, Pascale Marangé, Marie Duflot-Kremer, Jean-François Pétin

Abstract Industry 4.0 implies new scheduling problems linked to the optimal using of flexible resources and to mass customisation of products. In this context, first research results show that Discrete Event Systems models and tools are a relevant alternative to the classical approaches for modelling scheduling problems and for solving them. Moreover, the challenges of the industry 4.0 mean taking into account the uncertainties linked to the mass customisation (volume and mix of the demand) but also to the states of the resources (failures, operation durations, ...). The goal of this paper is to show how it is possible to use the simulation based on statistical model checking for taking into account these uncertainties and for evaluating the robustness of a given schedule.

Introduction

After the mechanisation, started in the middle of the XVIIIth century, the mass production and the electrification, at the end of the XIXth and finally the usage of computers and automation, at the end of XXth, the concept of Industry 4.0 assumes that we are at the beginning of a fourth industrial revolution. This revolution is based on the massive digitalisation and the new Information and Communication Technologies for facilitating the emergence of "smart factories". However, its enforcement highlights new challenges dealing with the management and control of the produc-

Sara Himmiche, Alexis Aubry, Pascale Marangé, Jean-François Pétin
Université de Lorraine, CRAN, UMR7039, Campus Sciences, BP 70239, 54506 Vandœuvre-lès-Nancy Cedex, France e-mail: firstname.name@univ-lorraine.fr

Sara Himmiche, Alexis Aubry, Pascale Marangé, Jean-François Pétin
CNRS, CRAN, UMR 7039, France

Marie Duflot-Kremer
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France e-mail: marie.duflot-kremer@loria.fr

tion systems. In fact, this new paradigm offers, of course, new possibilities: flexibility and agility of production resources, communication capabilities of the resources between them and with their environment, local capability of decision (by the resource and/or by the product). However, the usage of these new capabilities leads to new issues like the mass customisation of the products and needs new control strategies of the production.

These new abilities imply also that the production systems of the future will have as intrinsic characteristics: uncertainties on the demand (volume and mix) or even on their fabrication recipes and a need of optimal using of the resources flexibility. To make it possible, it is necessary to challenge the classical control strategies, which often assume that the predicted demand is certain, that the manufacturing routes are perfectly defined and that resources are always available for computing a centralised production schedule – a global allocation of the production operations to the resources and definition of the starting and completion dates of the operations by satisfying the constraints of the considered system.

In this context, the classical approaches for scheduling, mainly based on operations research (mathematical programming, metaheuristics. . .) become less efficient because they often consider a stable environment without taking into account the dynamics of the system and its perturbations.

The community of Discrete Event Systems (DES) precisely studies concepts for modelling systems with real time and random aspects, offering a modelling and analysing power that is appreciable. The methods and languages based on the DES theory became in fact a realistic alternative to classical methods for scheduling the production [10, 14].

The objective of this article is to present an approach based on DES models and tools for evaluating the robustness of a production schedule subject to uncertainties on the operations durations on the machines.

The remaining of the article is organised in five sections. The next section is dedicated to the presentation of the production scheduling problem that we want to address. In the second section, the concept of robustness is developed and a robustness definition is proposed for dealing with perturbations that are modelled by stochastic data. The proposed approach for evaluating the robustness level and the DES models that supports this approach are presented in the third section. An academic example is used in the fourth section for illustrating and discussing the approach. Conclusions and Perspectives are given in the last section.

1 Definition of the problematic

1.1 The general scheduling problem

Classically, scheduling the production consists in (i) allocating the production operations to resources and (ii) sequencing the operations on these resources (defining

the order of the operations), also satisfying the constraints defined by the considered production system and optimising a criterion (the total duration, the number of late operations...). Moreover, when the duration of each operation is considered as perfectly known and static, then it is possible to fix the starting and completion dates of each operation.

The most common criterion is the total duration of the schedule (the makespan, classically denoted as C_{max}). Among the common constraints are the precedence constraints that fix the sequence of operations for each product route.

Regarding its definition, the scheduling problem can be considered as an optimisation problem. Thus, the classical operational research tools (mathematical programming, metaheuristics...) have for a long time been the only tools used for solving these problems.

The classical approach for solving this type of problem is the predictive approach: an *off-line* algorithm – the predictive algorithm – computes an optimal schedule S_I^* for a predicted scenario I (fixing the value of each input parameter of the problem), considered as certain and static, and guarantees a local performance measured by a criterion z and valued by z_I^* on I. The performance z_I^* is only guaranteed for this scenario. However, the production system is naturally submitted to perturbations, and thus, the schedule S_I^* is actually applied to a realised scenario I' that is different from the predicted scenario I. Eventually, the application of the schedule S_I^* to I' can lead to constraints violations, making the schedule no more feasible for I'. Moreover, the real measured performance $z(S_I^*, I')$ can be really far from the predicted one z_I^*.

The limits of the classical approach are clearly highlighted here: it does not guarantee any performance if the realised scenario gets away from the predicted scenario. The need to take into account the perturbations and to propose an approach that is able to compute schedules with good performances despite this perturbations is thus a critical issue.

1.2 Scheduling flexible manufacturing systems

A workshop is defined by a set J of products that have to be processed on a set M of machines (the number of machines is given by card(M) and is denoted as $\mathscr{M}$). Each product j has to follow a production route O_j^J that defines a set of operations to be executed for processing the product j. The execution of the operation o_{jk} (k^{th} operation of the route O_j^J) needs a machine m that must be qualified for this operation, occupying this machine during d_{jkm} time units.

The global set of operations available in the workshop is given by $O^J = \bigcup_{j \in J} O_j^J$ and the total number of operations in the workshop is thus given by card(O^J) and is denoted as $\mathscr{O}$).

There exist different types of classical workshops depending of the possible flows of products: Flow-Shop, Job-shop, Open-shop. According to the type of workshop,

some precedence constraints between the operations are existing, defined by the routes O_j^j.

Regarding the criterion z, we will consider only the minimisation of the total duration of the schedule (classically denoted as C_{max}). That means that $z = C_{max}(S,I)$ and assesses the total duration of the schedule S on the scenario I (that fixes the duration of each operation on the machines).

1.3 The uncertainties

The execution duration of an operation o_{jk} on a machine m for which it is qualified can be rarely known with certainty. A reference duration d_{jkm}^{ref} can often be given: it corresponds to the duration given by the methods engineers. In practice, we will consider that the real execution duration is a stochastic data following a probabilistic distribution with expected value d_{jkm}^{ref}.

On the contrary, the routes O_j^j are supposed to be perfectly known and without variation and the machines are supposed to be without failures.

2 Robustness in scheduling

Tackling uncertainty in scheduling is not a new problem. We can distinguish two strategies when dealing with perturbations in scheduling that are complementary:

- Trying to take into account the perturbations - proactively - before the production for building a robust schedule
- Build an on-line schedule taking into account the real state of the system and the real input parameters

Robust scheduling consists in - according to predefined perturbations (meaning that perturbations are seen here as deterministic uncertainty) and their model - building a schedule that is able to guarantee some performances despite the modelled uncertainties. Performances that must be guaranteed clearly depends on the considered problem. But as an example, these performances could be a deadline to be respected despite uncertainty on the operations duration.

In the first chapter of [5], the robustness is defined as follows: *a schedule is robust if its performance is rather insensitive to the data uncertainties*. This definition, even if it has the merit to make the consensus, remains poorly precise insofar as it needs to define what means “to be rather sensitive to”. That means that some metrics for characterising this sensitiveness are necessary.

In the case where the uncertain data are modelled as random variables (possibly in given intervals), we propose to use the following robustness definition.

Definition 1. A schedule S is conditional-robust face to uncertainties modelled by a vector of random variables I if its probability of satisfying a robustness condition is higher than a given threshold P_{lim}. This can be mathematically formalised by the following condition:

$$Pr(z(S,I) \leq L_\lambda) \geq P_{lim} \tag{1}$$

In this definition, $z(S,I) \leq L_\lambda$ is the robustness condition that is expected to be satisfied despite the uncertainties on I. Typically, L_λ and P_{lim} must be defined by the decision maker who first defines L_λ as the expected performance and then P_{lim} as the minimal desired probability for satisfying the robustness condition. In the context of our problem, as defined previously, $z(S,I) = C_{max}(S,I)$. z assesses the global duration of the schedule S for the scenario $I = \{d_{jkm}\}_{jkm}$ that is a vector of random durations d_{jkm} for the operation o_{jk} on the different machines. Moreover, it is assumed that it is necessary that the global duration of the schedule does not exceed a given deadline $\tilde{d}$ despite these uncertainties. The robustness definition 1 is then true if the robustness condition above has a probability higher than a given probability P_{lim} when the durations d_{jkm} follow a given probability distribution. This can be expressed by the following condition: $Pr(C_{max}(S,I) \leq \tilde{d}) \geq P_{lim}$.

3 A DES-based approach for evaluating the robustness

The objective of this section is to show how models based on Stochastic Timed Automata and associated Statistical Model Checking can be used for evaluating the robustness level of a given schedule (definition 1), and how it is possible to use these models for helping the decision maker to decide between several schedules.

3.1 Approach overview

The proposed approach includes three steps (see figure 1).

- The first step consists in computing a deterministic schedule taking into account the information of the workshop (qualifications of the machines, reference operations durations on the machines, routes of the products) that are here considered as certain and static. The computation of this schedule can be done manually by an expert, or using a classical approach (metaheuristics, Mixed-Integer Linear Programming ...), or even using DES models and tools as in [3, 14, 17, 13]. This step is not addressed in this paper.
- The second step consists, on the schedule obtained after executing step 1, and considering uncertainties on the operations durations, to evaluate the robustness level determined by $Pr(C_{max}(S,I) \leq \tilde{d})$ with $I = \{d_{jkm}\}_{jkm}$ the random vector of operations durations. The deadline $\tilde{d}$ is a reasonable time limit fixed by the decision maker. Its value can be fixed according to a reference duration that can be

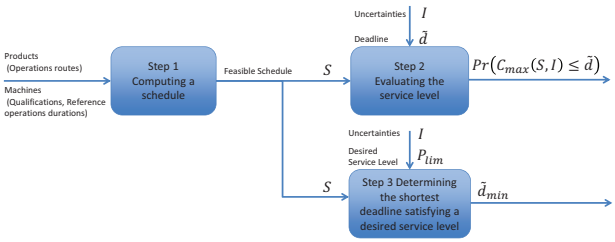

Fig. 1 Procedure for evaluating the robustness of a schedule

the makespan of the evaluated schedule on a reference scenario $Iref = \{d^{ref}_{jkm}\}_{jkm}$ denoted by C^{ref}_{max} (we accept to get away from a reference value but without exceeding $X\%$) or can be an arbitrary value corresponding to the horizon of the schedule for instance.

- The third step consists, for the initial schedule of step 1, for the same uncertainties as in step 2, and for a given desired robustness level P_{lim}, in determining the shortest deadline $\tilde{d}_{min}$ that satisfies $Pr(C_{max}(S,I) \leq \tilde{d}_{min}) \geq P_{lim}$.

3.2 Models

In order to implement the procedure presented in the previous section, it is necessary to have models that can represent: the different states of the resources, the operations routes for the products, the allocation and sequencing of the operations according to the given schedule, the uncertainties on the operations durations. Moreover, it will be necessary to find a way for evaluating the probability to satisfy some properties in order to assess, at the end, the robustness level of the schedule.

There exist several DES tools for satisfying such modelling requirements. We can cite Stochastic Automata [12, 6], Stochastic Petri Nets [2, 11] and Stochastic Automata Networks [15, 16].

3.2.1 Stochastic Timed Automata

Stochastic Timed Automata (STA) [7] are derived from the class of Timed Automata defined in [1]. Informally, a timed automaton as used in UppAal [4] is a model with a finite number of locations, and a finite number of real valued clocks. Discrete and instantaneous transitions can lead from one location to another. Those transitions can be equipped with guards (conditions on clocks that need to be enabled to fire the transition), resets (a subset of clocks whose value will be set to zero on firing the transition) and synchronisation labels (in order to synchronise two automata together). They can also include integer valued variables in guards and updates.

The STA give a probabilistic semantics to TA by probabilistically resolving the deterministic choices in the system, using probabilistic choice between several enabled transitions, and probabilistic distributions (uniform or exponential) to chose the delay before firing the next transition. This model is precisely the one implemented in the statistical model checker UppAal SMC. In the following, we will describe our models using UppAal SMC syntax.

In the following of the article, the localities will be noted in bold case, the events will noted in italic case, the guards and the invariants will be noted in [] and the updates will be noted in bold case and underlined.

3.2.2 STA models for a schedule

In order to evaluate the robustness of a given schedule, it is first needed to model this schedule. Its characteristics are:

- an allocation of each operation to a unique machine,
- the sequencing of the operations on the machines

Moreover, this schedule has to be admissible, *i.e.* to satisfy the precedence constraints defined by the routes of the products. In order to model a schedule, two generic models (model patterns) have been defined. According to the solved problem, these patterns are instantiated as many times as necessary.

The first model given in figure 2 is an instantiation of the operation pattern and represents the evolution of the states of the operation according to the route of the product it is linked to, and according to the sequencing in the machine allocated to this operation. The operation pattern is based on previous works [9, 13]. We need a copy of this pattern for each operation of each product. The behaviour of the model is as follows.

- In the locality ***Waiting_to_be_performed***, the operation is waiting until its enabling conditions (in green) are satisfied. The model can evolve when the previous operation in the route of the product and the previous operation in the sequence on the machine (defined by the schedule) have been completed and if the product is available (not undergoing another operation).
- If this guard is satisfied then the operation sends a request (requested[ordo[i].M]) to the machine allocated, according to the schedule, for executing this operation. Moreover, the status of the product is set to unavailable and the operation now waits to be completely executed in the locality ***Waiting_to_be_executed***.
- Upon receiving an event (executed[i]) telling that the machine has completed the operation, the model moves to the final locality ***End_of_opeation***, and updates variables in order to show that the product is available and the operation has been completed.

The second model given in figure 3 is an instantiation of the machine pattern and represents the behaviour of a machine when executing an operation.

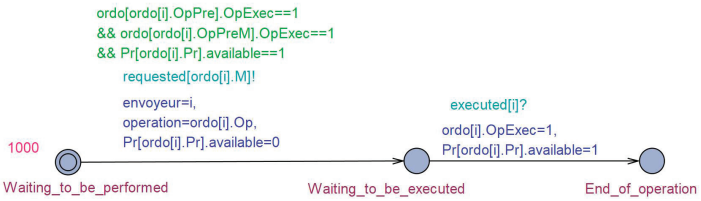

Fig. 2 Instantiation of an operation pattern using UppAal

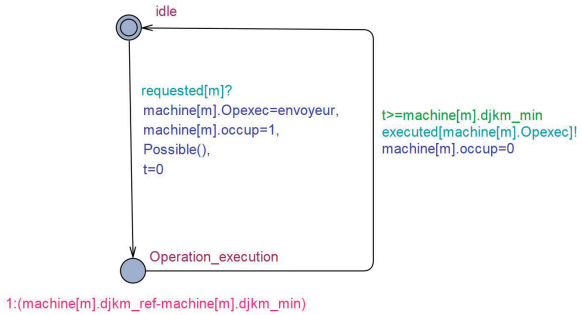

Fig. 3 Model pattern β^m for the machine m

The machine pattern has to be instantiated for each machine. It evolves as follows:

- In the locality ***idle***, the machine m is waiting for the request requested[m], which means that an operation is asking to be executed. As the machine is available, it accepts the operation, set its status to occupied, stores the id of the operation, resets the local clock to compute the execution time.
- The model moves towards the locality ***Operation_execution***, and remains there until the operation is completed, which takes a time exponentially distributed which parameter $\lambda = \frac{1}{d^{ref}_{jkm} - d^{min}_{jkm}}$ where d^{min}_{jkm} is the minimal duration of the operation o_{jk} on the machine m.

3.3 Evaluation of the robustness

Based on the previous models and using the model checker UppAal, the steps 2 and 3 of the figure 1 are implemented.

Concerning the verification of probabilistic systems, two families of model checkers can be distinguished. So called (numerical) probabilistic model-checking

consists in computing, as precisely as one wants, the exact value of a probability, by using numerical methods based on matrix calculation. The statistical model-checking is based on the sampling of executions in order to estimate a probability by using Monte-Carlo style methods.

The main advantage of the first method is its precision regarding the obtained results (without the risks linked to the usage of statistical methods). However, the second method avoids the combinatory explosion because it does not need to build the complete state space of the system. Moreover, it permits to have richer models (regarding the evolution of the variables or the probability distributions).

In order to answer to the initial issue, *i.e.* taking into account both time aspects inherent to the scheduling problem and the variability of the system, a tool able to catch deterministic timed activities but also able to introduce some stochastic aspects in other delays (as the treatment of the operations durations) was needed. Regarding these aspects, it was natural to select UppAal SMC [7], a tool that accepts the timed models initially verified by UppAal, but also designed for adding a probabilistic semantics to timed models.

For the robustness level (step 2 of the procedure), the question asked to the model checker is: What is the probability, within deadline $\tilde{d}$ to have completed all the operations.

$$Pr[\leq \tilde{d}](<> forall(i : int[1, NbOp-1])Ordo[i].OpExec == 1) \tag{2}$$

For step 3, the idea is to find the smallest value d_{min} for which the probability to complete all the operation within d_{min} time units is higher than a fixed value P_{lim}. That means computing the smallest value of $\tilde{d}$ for which the following property is satisfied:

$$Pr[\leq \tilde{d}](<> forall(i : int[1, NbOp-1])Ordo[i].OpExec) \geq P_{lim} \tag{3}$$

In order to compute this value, a binary search algorithm can be used. The idea is to start with an interval of the possible values of d_{min} and to reduce iteratively this interval until a sufficient precision for the value of d_{min} is reached. This algorithm consists in:

1. defining a first interval $[d_{inf}; d_{sup}]$ in which the value of d_{min} is guaranteed to be. d_{inf} can be fixed to C^{ref}_{max} and d_{sup} can be fixed to the sum of the durations of all the operations,
2. reducing the size of the interval $[d_{inf}; d_{sup}]$ by picking the middle value, checking the property (3) with this middle value ($\tilde{d} = \frac{d_{inf}+d_{sup}}{2}$),
3. if the property is satisfied, updating the interval $[d_{inf}; d_{sup}]$ with $[d_{inf}; \frac{d_{inf}+d_{sup}}{2}]$, else updating the interval with $[\frac{d_{inf}+d_{sup}}{2}; d_{sup}]$,
4. going back to the step 2 of the algorithm with this updated interval (until a given precision is reached).

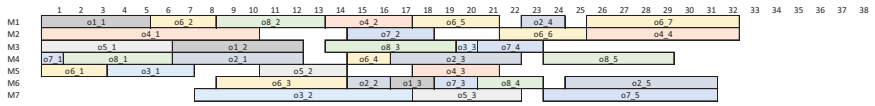

Fig. 4 Possible schedule in 32 time units

4 Experiments

The approach, presented in the previous section, has been experimented on an academic example taken from [8] and adapted for taking into account the uncertainties on the operations durations. The objective of this example is to schedule eight products. The workshop contains a CNC lathe (M1), a grinding machine (M2), three milling machines (M3 to M5) and two furnaces (M6 and M7). The characteristics of the products to be completed and the workshop are detailed in the table 1.

Product	Operation	Ressource	d^{min}_{jkm}	d^{ref}_{jkm}	Product	Operation	Resource	d^{min}_{jkm}	d^{ref}_{jkm}
1	$o1\_1$	CNC lathe	4	5	6	$o6\_1$	milling machine	2	3
1	$o1\_2$	milling machine	4	6	6	$o6\_2$	CNC lathe	2	3
1	$o1\_3$	furnace	1	2	6	$o6\_3$	furnace	4	6
2	$o2\_1$	milling machine	4	6	6	$o6\_4$	milling machine	1	2
2	$o2\_2$	furnace	1	2	6	$o6\_5$	CNC lathe	3	4
2	$o2\_3$	milling machine	4	6	6	$o6\_6$	grinding machine	3	4
2	$o2\_4$	CNC lathe	1	2	6	$o6\_7$	CNC lathe	5	7
2	$o2\_5$	furnace	5	7	7	$o7\_1$	milling machine	1	1
3	$o3\_1$	milling machine	3	4	7	$o7\_2$	grinding machine	3	4
3	$o3\_2$	furnace	8	10	7	$o7\_3$	furnace	1	2
3	$o3\_3$	milling machine	1	1	7	$o7\_4$	milling machine	2	3
4	$o4\_1$	grinding machine	4	6	7	$o7\_5$	furnace	6	8
4	$o4\_2$	CNC lathe	1	2	8	$o8\_1$	milling machine	4	5
4	$o4\_3$	milling machine	4	6	8	$o8\_2$	CNC lathe	4	5
4	$o4\_4$	grinding machine	1	2	8	$o8\_3$	milling machine	4	6
5	$o5\_1$	milling machine	4	5	8	$o8\_4$	furnace	2	3
5	$o5\_2$	grinding machine	4	6	8	$o8\_5$	milling machine	4	6
5	$o5\_3$	furnace	1	2					

Table 1 Route of the products and characteristics of the machines

Using the method given in [13] and considering a reference operation duration d^{ref}_{jkm} (given in the table 1), two schedules can be obtained for instance. They are represented in the figures 4 and 5. Their respective global duration have the same magnitude. In order to help the decision maker to decide between these two schedules, the one to be executed in the workshop, the approach presented in the previous section has been used for each schedule. The robustness level has been first evaluated for each schedule considering that the deadline $\tilde{d}$ has been fixed to 110% of the reference duration fixed by $C_{max}(S, I^{ref})$: 35 for the first schedule and 42 for the second one. That means the decision maker accepts that the duration of the schedule can increase of 10%, but no more.

In order to evaluate the robustness level of the two schedules according to the desired deadline $\tilde{d}$, the following property has been tested on UppAal SMC:

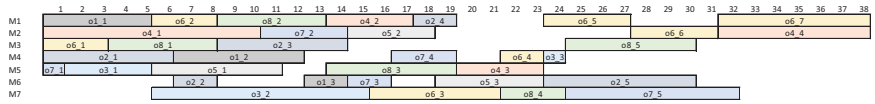

Fig. 5 Possible schedules in 38 time units

$$Pr[<=35(resp.42)](<>forall(i:int[1,NbOp-1])Ordo[i].OpExec==1) \quad (4)$$

For the first schedule, the obtained probability is 25% and for the second schedule it is 50%. That means that, despite the uncertainties on the operations duration, when executing the second schedule, the products have a probability of 50% to be completed before the desired deadline (42 units of times).

In order to evaluate the smallest deadline that allows a robustness level of 85%, a binary search algorithm has been executing by iteratively test the following property UppAll SMC:

$$Pr[\leq date\_accept](<>forall(i:int[1,NbOp-1])Op(i).End\_operation\_o_{jk}) \geq 0,85 \quad (5)$$

For the first schedule, the obtained smallest acceptable deadline is 42 time units and for the second schedule the obtained smallest acceptable deadline is 47 time units. That means that the decision maker has to accept to increase the deadline to 47 time units for obtaining a robustness level of 85%.

Even though the second schedule ends in 38 time units (later than the first one), the first property shows that it is more robust when considering the uncertainties on operations duration. This conclusion is aligned with previous works that show that the most robust schedule is rarely the optimal one (compromise between optimality and robustness).

5 Conclusion

This paper proposed an approach based on Statistical Model-Checking of Stochastic Timed Automata (STA) for evaluating the robustness of one schedule face to uncertainties on the operations durations. This approach first proposes some modelling patterns in STA for modelling a schedule and then proposed to use the Statistical Model-Checking for formally evaluating the robustness level of the modelled schedule. The models that have been proposed are independent of the type of the workshop to be scheduled. This makes the approach an interesting alternative to classical approaches that are often dedicated to the scheduling problem.

In the future, this approach must be extended and enriched for taking into account other types of perturbations (the failures for instance) and other probability distributions. This approach starts with a given schedule to be evaluated. The next

step is to propose a method for computing a robust schedule from scratch according to the modelled perturbations and to a desired robustness level.

References

1. R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
2. P. Ballarini, H. Djafri, M. Duflot, S. Haddad, and N. Pekergin. Petri nets compositional modelling and verification of flexible manufacturing systems. *Proceeding of the 7th Conference on Automation Science and Engineering*, 2011.
3. G. Behrmann, E. Brinksma, M. Hendriks, and A. Mader. Production scheduling by reachability analysis - a case study. *International Parallel and Distributed Processind Symposium*, 2005.
4. G. Behrmann, A. David, and K.G. Larsen. A tutorial on uppaal. In *Proc. Formal Methods for the Design of Real-Time Systems*, number 3185 in Lecture Notes in Computer Science, pages 200–236. Springer, 2004.
5. Jean-Charles Billaut, Aziz Moukrim, and Eric Sanlaville. *Flexibility and Robustness in Scheduling*. ISTE, 2010.
6. C. G. Cassandras and S. Lafortune. *Introduction to discrete event systems, second edition*. Springer, 2008.
7. Alexandre David, Kim G Larsen, Axel Legay, Marius Mikučionis, and Danny Bøgsted Poulsen. Uppaal smc tutorial. *International Journal on Software Tools for Technology Transfer*, 17(4):397–415, 2015.
8. Vincent Giard. *Gestion de la production et des flux: avec CD livre électronique+ Logiciels+ Animations*. Economica, 2003.
9. Sara Himmiche, Alexis Aubry, Pascale Marangé, and Jean-François Pétin. Modeling Flexible Workshops Scheduling problems: evaluating a Timed Automata based approach vs MILP. In *20th World Congress of the International Federation of Automatic Control, accepted*, Toulouse, France, 2017.
10. A. Kobetski and M. Fabian. Time-optimal coordination of flexible manufacturing systems using deterministic finite automata and mixed integer linear programming. *Journal of Discrete-Event Dynamic Systems: Theory and Applications*, 19(3):287–315, 2009.
11. P.S. Kritzinger and F. Bause. *Stochastic Petri nets - An Introduction to the theory*. Vieweg+Teubner, 2002.
12. M. Kwiatkowska, G. Norman, D. Parker, and J. Sproston. Performance analysis of probabilistic timed automata using digital clocks. *Formal Methods in System Design*, 2006.
13. Pascale Marangé, Alexis Aubry, and Jean-François Pétin. Ordonnancement d'ateliers à partir de patrons de modélisation basés sur des automates communicants. In *11th International Conference on Modeling, Optimization and Simulation*, Montréal, Canada, August 2016.
14. S. Panek, S. Engell, and O. Stursberg. Scheduling and planning with timed automata. In *16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering*, pages 1973?–1978. Elsevier, 2006.
15. B. Plateau and K. Atif. Stochastic automata network of modeling parallel systems. *IEEE Transactions on Software Engineering*, 17(10):1093–1108, Oct 1991.
16. W. J. Stewart, K. Atif, and B. Plateau. The numerical solution of stochastic automata networks. *European Journal of Operational Research*, 1995.
17. S. Subbiah and S. Engell. Short-term scheduling of multi-product batch plants with sequence-dependent changeovers using timed automata models. In *20th European Symposium on Computer Aided Process Engineering*, number 28, pages 1201–1206. Elsevier, 2010.